Installation and User Guide for CO_2DAS Data Ingestion System: A System for Real-time Large-Scale Parallel Data Ingestion

PUORG Research Group
The University of Michigan
June 30, 2011

Abstract

The CO_2DAS Data Ingestion System (DIS) is responsible for discovering and remembering sources of remote data, and ensuring that new data appearing at a source are staged to local disk for storage and subsequent processing.

A prototype was built over the summer of 2011, comprising the Acquisition, Formatting, Transform, and Source Control components of the system. The prototype is currently in production, running on the Center for Advanced Computing's Flux cluster at the University of Michigan. This document provides the Installation and User's Guides for the prototype.

Table of Contents

PUORG Research Group The University of Michigan June 30, 2011 ..... **1**

Abstract **1**

1. Introduction **3**

2. Installation Guide **5**

2.1 Overview *5*

2.2 Resources *5*

2.3 Installation *5*

Get the Code 5

Unpack the Code 6

Code Hierarchy 6

Install the Code 6

3 User Manual **7**

3.1 Overview *7*

3.2 Create source specification file *7*

3.3 Start the DIS prototype *10*

3.4 Stop the DIS prototype *10*

3.5 Obtain the status of the DIS prototype *10*

3.6 Reset the DIS prototype *11*

3.7 Notes *11*

3.8 Diagnostics *11*

Parser 11

status-co2das 11

References **13**

Appendix A **14**

1. Introduction

The CO_2DAS Data Ingestion System (DIS) is responsible for discovering and remembering sources of data, and ensuring that new data appearing at a source are staged to local disk for storage and subsequent processing *[Michalak et al, 2010]*. Its design document describes the full acquisition system design *[Antonelli, 2011]*. A data flow diagram of the full system is shown in Figure 1.

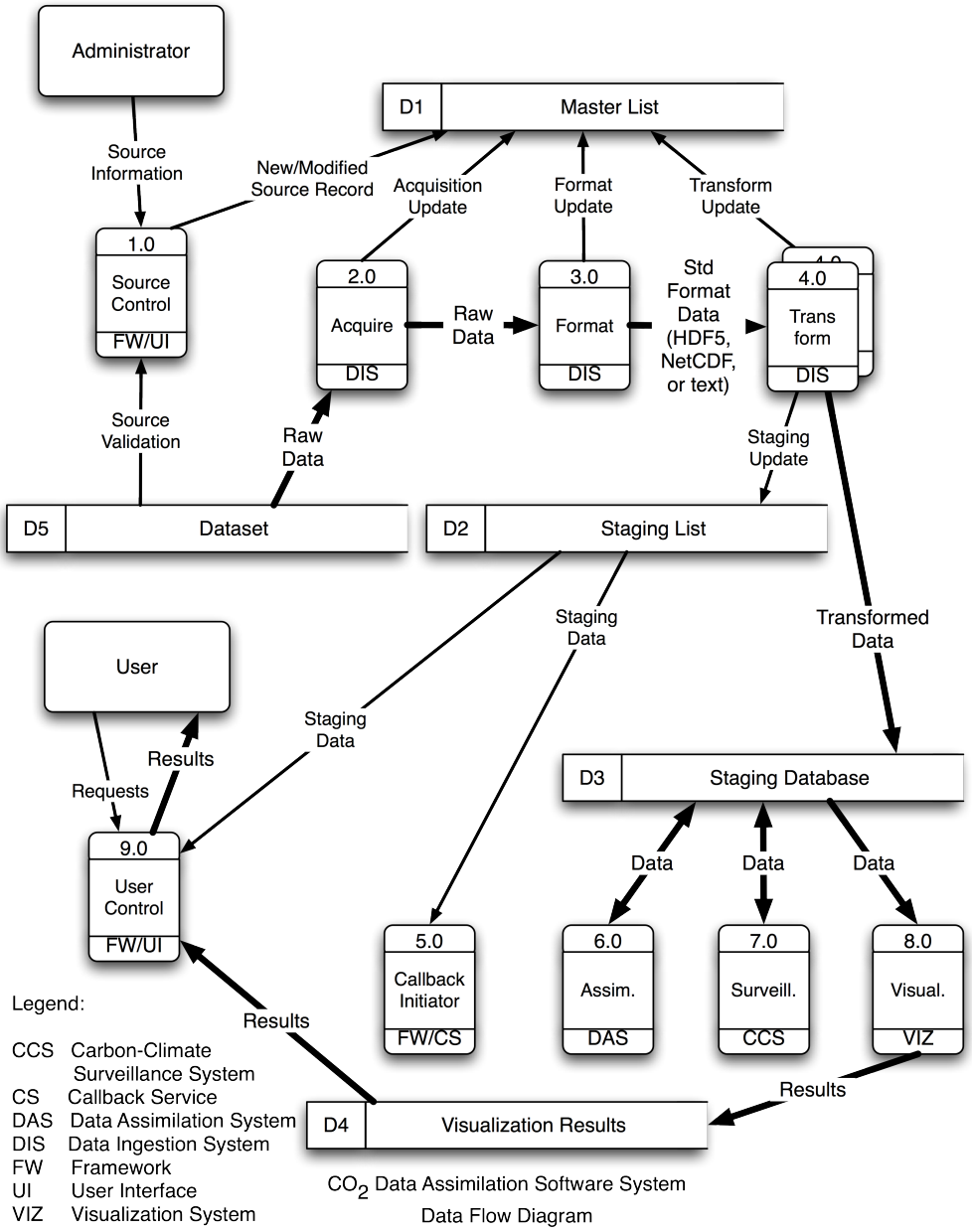

Figure 1

A prototype was built over the summer of 2011, comprising the Acquisition, Formatting, Transform, and Source Control components of the system. The prototype is currently in production on the Center for Advanced Computing's Flux cluster at the University of Michigan. This document provides the Installation and User's Guides for the prototype.

2. Installation Guide

2.1 Overview

The CO_2DAS Data Ingestion System prototype (DIS) is a suite of several C programs and bash scripts that are executed on a Linux platform. We have tested the suite on RHEL 5.6.

2.2 Resources

DIS requires the following resources in order to produce correct and timely results:

Computing Platform

DIS expects to run on in a computing cluster environment using the PBS Portable Batch System. We use the Torque Resource Manager fork of the Open PBS implementation and the MOAB Cluster Scheduling Suite, running on the Flux cluster provided by the Center for Advanced Computing at the University of Michigan. Compute nodes should possess at least 1 GB RAM.

We have also briefly tested DIS on a Torque Resource manager implementation (that is, without MOAB, which is not a free product, instead running with Torque's built-in cluster scheduler), running on a small cluster of two machines with 16 cores each,48 and 72 GB RAM, respectively, and 1 TB of disk. Torque is freely available at http://www.clusterresources.com/ .

Disk Storage

DIS expects at least several TB of available disk storage accessible on the cluster to maintain the CO_2DAS data cache. With many data sources, several dozen to several hundred TB will allow more data to be cached for longer periods of time.

Network Connectivity

For best performance, the compute nodes running DIS should have direct access to the internet for downloading of data. If this is not possible, DIS can acquire data by funneling downloads through the head node(s), but this may become a performance bottleneck.

2.3 Installation

Use the following procedure to install the DIS prototype. Running DIS will be covered in the next section.

Get the Code

You may download the DIS prototype code from our web site at http://www.puorg.umich.edu. You will need to provide login credentials to download from our site. Please see our web site for guidance on creating these credentials.

Unpack the Code

Unpack the DIS prototype code archive you selected into an arbitrary directory on your cluster head node. Nothing in the DIS package requires root authority, so you should install and run the code using a normal, unprivileged user account.

Code Hierarchy

The complete DIS prototype directory tree should be as follows:

```
si2:
bin  locks  logs  scripts  src  stats

si2/bin:
reset-co2das  restart-co2das  status-co2das  stop-co2das

si2/locks:

si2/logs:

si2/scripts:
coarsen_one_merra.sh  extract_one_merra.sh  fetchi-exec
extract_one_merra.gs  fetchi                sources.txt

si2/src:
Build.coarsenMetData  emitc     mapa2a.f
coarsenMetData.f      Makefile  parser-co2das.c

si2/src/emitc:
ctime.c  emitc.c  Makefile  README  smbgetdate.y
```

Install the Code

Follow the following steps to install the prototype:

1. Place the DIS bin directory in your PATH:

   ```
   set PATH=$PATH:your/install/dir/si2/bin
   ```

   You should place this directive into your `.bashrc` or similar file as well.

2. Enter the src directory and compile and install the code therein:

   ```
   cd si2/src
   ./Build.binaries
   ```

   Note: if your compile environment does not support `gfortran`, substitute the correct compiler invocation in `./Build.coarsenMetData` and rerun that script.

3. Update the parameters at the beginning of each of the following scripts, changing the string `/home/cja/work` to that to of your installation directory:

   ```
   si2/scripts/coarsen_one_merra.sh
   si2/scripts/extract_one_merra.sh
   ```

```
si2/scripts/fetchi
si2/scripts/fetchi-exec
si2/bin/reset-co2das
si2/bin/restart-co2das
si2/bin/status-co2das
```

4. Determine how often you wish DIS to attempt to fetch the next group of data from the sources. As shipped, DIS will schedule ingestion once per day. To change this value, edit `si2/scripts/fetchi` and change the parameter `D` to the desired number of hours between ingestion cycles. Note: D must be less than or equal to 24 hours.

5. Change these PBS job parameter comments in the fetchi and fetchi-exec scripts to match those required by your cluster operator:

```
#PBS -q flux
#PBS -l qos=cja_flux
#PBS -A cja_flux
```

3 User Manual

3.1 Overview

DIS uses a *source specification file* to determine sources of data, what data to download from the sources, and how to recode them once they are downloaded. Guided by this file, DIS will access the data sources, identify the files to download, perform the download, and post-process the data after downloading. The Master List shown in Figure 1 is implemented in the prototype as the source specification file.

Data are left in a *cache directory* on the local cluster, called the Staging Database in Figure 1. These data are then further processed by other components of CO_2DAS, and then eventually deleted or retained. These components are not covered further here.

3.2 Create source specification file

The source specification file is composed of groups of attribute-value paragraphs, one paragraph per source. A separate record defines each separate source, even if several individual source directories are located at the same URL.

The file is named `sources.txt` and is stored in the `si2/scripts` directory. The possible attributes and their meanings are shown in Table 1.

Attribute	Meaning	Mandatory?
record	Record number	yes
label	Source label (short, alphanumeric only)	yes
depends	List of labels on which this source depends; those sources will be acquired before this acquisition is started.	
title	Long source title	
documentation	URL of applicable data source documentation	
updatereate	Not used	
lastupdate	Not used	
site	The URL of the source	yes
root	The path to the root of the dat a tree at the source	yes
dir	The directory specification that will be used to retrieve the data files **(Note 1)**	yes
sample	A sample full pathname to a particular file	
begindate	Data will be acquired starting with this date.	yes
enddate	… and ending with this date	yes
duration	How many days of data are to be acquired and processed as a group	yes
format	The name of a script to format the data **(Note 2)**	
transform	The name of a script to transform the data **(Note 3)**	
group	The name of a script to operate on the data as a group, once all data in the group have been acquired	

Table 1

The sample specification file included in the distribution may be used as a model. A record from this file and the corresponding generated acquisition script is shown in Appendix A.

Note 1: The dir attribute may contain wildcards. These wildcards are expanded by the acquisition agent (using `parser-co2das`) to generate the actual filenames between the beginning and ending dates. The three most common wildcards are:

%Y	Four-digit year
%m	Two digit month, 01 through 12
%d	Two-digit day, 01 through 31

Many other wildcards may be used; see the `strftime(3)` Linux manual page for details by typing

```
$ man 3 strftime
```

Note 2: The user may supply the format script or use the script included with DIS. The choices for the format attribute value are:

`None`	No formatting of the downloaded data will be performed
`extract_one_merra.sh`	This formats the data as required by PCTM [Dunigan 2011].
`userscript`	This invokes a user-written script, here named "userscript". Arguments provided to the script by DIS include the year, month, day, and cached file name.

The specified script will be invoked immediately after each file is acquired by DIS. User scripts should be placed in the `si2/scripts` directory (and made executable) before DIS is started or restarted.

Note 3: The user may supply the transform script or use the script included with DIS. The choices for the transform attribute value are:

`none`	No transformation of the downloaded data will be performed
`extract_one_merra.sh`	This coarsens the data produced by `extract_one_merra.sh` to the granularity required by PCTM
`userscript`	This invokes a user-written script, here named "userscript". Arguments provided to the script by DIS include the year, month, day, and cached file name.

The specified script will be invoked immediately after each file is acquired by DIS. User scripts should be placed in the `si2/scripts` directory (and made executable) before DIS is started or restarted.

Note 4. The user will usually want to supply a group script to process a set of acquired files as a group after all files in the group have been acquired.. The choices for the group attribute value are:

`none`	No transformation of the downloaded data will be performed
`userscript`	This invokes a user-written script, here named "userscript". Arguments provided to the script by DIS include the year, month, day, and the names of all the files in the group.

The specified script will be invoked after all files in the group have been acquired by DIS. User scripts should be placed in the `si2/scripts` directory (and made executable) before DIS is started or restarted.

3.3 Start the DIS prototype

Specify

```
restart-co2das
```

to start DIS. The DIS fetch initiator (`fetchi`) will schedule one fetch agent (`fetchi-exec`) for immediate execution for each source found in the source specification file, all of which will run in parallel, as well as a new instance of itself to run at the specified time in the future. Each fetch agent will download the data and apply the specified format, transform, and group operations on the data. This process will repeat itself indefinitely until manually stopped.

3.4 Stop the DIS prototype

To stop the prototype, specify:

```
stop_co2das
```

This will delete the new instance of `fetchi` and stop future execution of DIS. Any current jobs running on behalf of the current cycle will be allowed to finish.

3.5 Obtain the status of the DIS prototype

Specifying

```
status-co2das
```

Will display the current status of the DIS, e.g.,

```
$ status-co2das
CO2DAS is using 383GB with 45TB of 143TB available in mds1:mds2:/nobackup.
Fetch initiator next runs at: Fri Jul 1 01:00:00 2011 EDT
```

This shows the current size of the CO_2DAS data cache and the amount of available space left in the cache filesystem for new data, as well as the time and date at which the fetch initiator will next run.

For a short time after a new instance of the fetch initiator is started, its time of execution is not available from the PBS batch system. In this case, DIS will display, e.g.,

```
Fetch initiator next runs at: 0100
```

indicating the hour (in 24-hour format) when DIS has requested the job to run, here 1 AM. (If this time is less than the current time, the job will run on the following day.)

3.6 Reset the DIS prototype

Specify

```
reset-co2das
```

to restart DIS as well as to cause all previously fetched and processed data to be re-processed. No cached data are deleted, nor are any duplicate data acquired again, and the reprocessed products simply replace any previous versions. If data previously stored at a data source are modified in place, this command is useful to acquire only the changed data, and then re-process all of the data to achieve a consistent re-processed product.

3.7 Notes

CO_2DAS fetches data continuously, formatting and transforming data as they are obtained. If a file cannot be fetched because it does not yet exist on a source, any formatting and transformation processing operations will fail silently. However, the group operation will not be invoked until all data in the group are present, and the fetch initiator's state will not be updated for that source. Thus the fetch attempts for that group will continue indefinitely until all data in the group can be downloaded, at which time all formatting, transformation, and group operations will be successfully processed.

If multiple file specification evaluations yield the same remote file, it will only be downloaded once per group.

3.8 Diagnostics

Parser

This diagnostic is output by the parser when generating the script file and may be found therein. These files have a `.script` suffix and are located in the si2/scripts directory.

```
# All data for the specified date range have been downloaded from this
source.
```

This means a fetch agent has determined that all data from the specified source in the specified date range have been downloaded to CO2DAS. More data can be fetched by updating the date range in the appropriate source record.

status-co2das

These diagnostics are output from `status-co2das:`

```
No fetch initiator queued ... get help
```

This response from status-co2das indicates no fetch initiator is queued for execution, which means that after the current cycle finishes, no more cycles can be started by CO2DAS. This can

be resolved by running restart-co2das. If this condition persists, it may indicate a fault in the batch queuing system.

`Multiple fetch initiators queued ... get help`

This response from status-co2das indicates more than one fetch initiator is queued for execution. As locking primitives sequence access to data sources, CO2DAS will continue to operate without loss of consistency. However, this condition will consume extra CPU and network resources, and will persist until one of the fetch initiators is deleted with the `qdel` command.

`I don't know when the fetch initiator runs next, get help`

This response indicates that `status-co2das` cannot determine whether or not a fetch initiator is queued or when it will execute. This may indicate a fault with the underlying batch system or with its tools, such as `qstat`.

References

C.J. Antonelli, "CO_2DAS Data Ingestion System Design, Version 1.0," PUORG Research Group, Department of Civil and Environmental Engineering, University of Michigan, April 2011.

Meehl, G. A., W. M. Washington, J. M. Arblaster, and A. X. Hu, "Factors affecting climate sensitivity in global coupled models," J. Clim., 17, 1584 – 1596, 2004..

A.M. Michalak et al, "SI2-SSI: Real-Time Large-Scale Parallel Intelligent CO_2 Data Assimilation System," 2010.

Appendix A

As an example, consider this record:

```
record:          3
label:           MERRA_Fx
depends:         MERRA_Fv,MERRA_Fe
title:           Modern Era Retrospective-analysis for Research And
Applications (MAT3FXCHM)
documentation:
http://mirador.gsfc.nasa.gov/collections/MAT3FXCHM__5.2.0.shtml
updaterate:
lastupdate:
site:            ftp://goldsmr1.sci.gsfc.nasa.gov/
root:            data/s4pa//MERRA/MAT3FXCHM.5.2.0/
dir:             %Y/%m/MERRA300.prod.assim.tavg3_2d_chm_Fx.%Y%m%d.hdf
sample:
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0//2009/01
/MERRA300.prod.assim.tavg3_2d_chm_Fx.20090101.hdf
begindate:       01/01/2007
enddate:         12/31/2007
duration:        8
format:          extract_one_merra.sh
transform:       coarsen_one_merra.sh
group:           none
```

Here the group size consists of eight days of data. During the first cycle after this record is added, DIS will acquire data for 1/1/2007 through 1/8/2007. For each day in the group, the wildcards of the `dir` attribute are expanded by `parser-co2das` to reflect the day being acquired, and the resulting file name is added to the script. The format and transform commands and their arguments, reflecting the day being acquired, are added to the script. Finally, if a group attribute is defined (not shown), the group script is invoked with the starting date for the group and all eight filenames. For example, PCTM, which requires eight days of data, could be invoked here.

After generating this script, DIS submits it as a batch job, and advances the acquisition to the next group, for 1/9/2007 through 1/16/2007. The next cycle will acquire these data, and this process continues until the end date of 12/31/2007 is reached.

Note: If the end date is reached before the end of the group is reached, additional days are acquired after the end date to complete the last group.

`parser-co2das` will generate the following script from this record:

```
#PBS -S /bin/sh
#PBS -N MERRA_Fx
#PBS -q flux
#PBS -l qos=cja_flux
#PBS -A cja_flux
#PBS -m n
#PBS -V
```

```
echo + Fri Jul  1 01:00:58 EDT 2011 FTCHB `hostname` "Modern Era
Retrospective-analysis for Research And Applications (MAT3FXCHM)"

ssh nyx-login-intel1 <<EOF
lockfile /home/cja/work/si2/locks/3.lock
echo + `date` LOCKD `hostname` /home/cja/work/si2/locks/3.lock
echo + /home/cja/work/si2/bin/parser-co2das -LMERRA_Fx -
Sftp://goldsmr1.sci.gsfc.nasa.gov/ -Rdata/s4pa//MERRA/MAT3FXCHM.5.2.0/ -
D%Y/%m/MERRA300.prod.assim.tavg3_2d_chm_Fx.%Y%m%d.hdf -
C/nobackup/cja/proto/cache -b01/01/2007 -e12/31/2007 -d8 -
Fextract_one_merra.sh -Tcoarsen_one_merra.sh -Gnone -
P/home/cja/work/si2/scripts
#!/bin/bash

# CO2DAS fetch script, version 1.1
# site=ftp://goldsmr1.sci.gsfc.nasa.gov
# root=data/s4pa//MERRA/MAT3FXCHM.5.2.0
# cache=/nobackup/cja/proto/cache

# wget argument:  -N: compare sizes & timestamps, don't fetch if identical
# wget argument:  -x: create local directory hierarchy to mirror remote's
# wget argument: -nv: log only names of fetched files

if ! cd /nobackup/cja/proto/cache; then echo unable to change to cache
directory /nobackup/cja/proto/cache ... abend; exit 1; fi

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070720.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 20
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070720.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 20
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070720.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070721.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 21
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070721.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 21
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070721.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070722.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 22
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070722.hdf
```

```
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 22
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070722.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070723.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 23
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070723.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 23
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070723.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070724.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 24
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070724.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 24
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070724.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070725.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 25
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070725.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 25
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070725.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070726.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 26
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070726.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 26
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070726.hdf

wget -N -x -nv
ftp://goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3FXCHM.5.2.0/2007/07/
MERRA300.prod.assim.tavg3_2d_chm_Fx.20070727.hdf
/home/cja/work/si2/scripts/extract_one_merra.sh 2007 07 27
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070727.hdf
/home/cja/work/si2/scripts/coarsen_one_merra.sh 2007 07 27
/nobackup/cja/proto/cache/goldsmr1.sci.gsfc.nasa.gov/data/s4pa//MERRA/MAT3
FXCHM.5.2.0/2007/07/MERRA300.prod.assim.tavg3_2d_chm_Fx.20070727.hdf
```

```
# All files in group downloaded -- this group will be processed.


exit 0
EOF

# release global source fetch lock
/bin/rm -f  /home/cja/work/si2/locks/3.lock
echo + `date` ULOCK `hostname` /home/cja/work/si2/locks/3.lock
```